

## **REMARKS**

### **I. The Independent Claims Have Been Amended to Distinguish Thurlo**

In the outstanding Office Action, the indication of allowability of the pending claims was withdrawn, and each of the independent claims was rejected as being anticipated by or obvious in view of a new reference: U.S. Patent No. 5,835,772 to Thurlo. Each of the independent claims recites an element relating to a memory device storing “program code.” In the Office Action, it was asserted that the driver in Thurlo, which provides a host device with various parameters and protocols needed for the host device to interface with a memory device, corresponds to the recited “program code.” However, in some of Applicants’ embodiments, “program code” is not a driver needed to interface with the memory device but rather is an application (such as, but not limited to, an image viewer, an audio player, a calendaring tool, a word processor, etc.), the relevant functions of which are recited in the respective claims. Accordingly, to clarify the claims and distinguish Thurlo, Applicants have amended the independent claims to specifically recite that “the program code is not a driver.”

Since Thurlo only teaches a driver and not the recited “program code,” Applicants respectfully request that the rejections of the independent claims and their dependent claims be withdrawn.

### **II. Additional Grounds of Patentability**

In addition to the “program code”/“driver” distinction discussed above, each of the independent claims recites additional elements that provide additional grounds of patentability.

**A. Independent Claims 3, 7, 15, 16, 88, and 89**

Independent Claims 3, 7, 15, 16, 88, and 89 each recite an element relating to reading data stored in a memory device using program code stored in the memory device. It was asserted that this element was shown in Thurlo. Applicants respectfully disagree.

Thurlo teaches a memory device that stores driver code, which allows a host device to communicate with the memory device. However, in the examples given in Thurlo, the memory device is blank, and data is *written* to the memory device — *not read* from the memory device, as recited in independent Claims 3, 7, 15, 16, 88, and 89. In one example in Thurlo, the memory device is used as a “roll of film” that a digital camera uses to *store* digital pictures. See col. 5, lines 7-8 (“After a picture is taken, the digital camera *stores* the data associated with the picture into the flash card” (emphasis added)). In another example in Thurlo, the memory device is used as a “blank tape” in an audio *recording* device to *store* an audio file. See col. 5, lines 16-18 (“For an alternate embodiment, the system is an audio *recording* device which *stores* digital audio signals in peripheral flash memory” (emphasis added)). See also col. 7, lines 18-22 (“Thus, in this manner, the interpretive driver code, originally resident on the peripheral device, has provided a software interface between the system and peripheral, thereby enabling communication for the execution of a *write* operation” (emphasis added)).

In summary, the examples in Thurlo only teach using the driver to *write* data to — *not read* data from — the memory device. This provides an additional ground of patentability.

**B. Independent Claims 15, 33, 37, 39, 88, and 91**

Independent Claims 15, 33, 37, 39, 88, and 91 each recite an element relating to the fact that the program code is operative to store data *only* in the memory device that stores the

program code. It was asserted that this element was shown in Thurlo. Applicants respectfully disagree.

In Thurlo, driver software (the asserted “program code”) in a memory device makes a host device aware of certain parameters and protocols associated with that *type* of memory device. See col. 2, lines 8-9 (“Different drivers are associated with the same *type* of peripheral device” (emphasis added)). Further, as shown in Figure 2, the parameters identified by the driver (e.g., speed, block size, memory size, etc.) are parameters that are common among memory devices of the same type. Accordingly, the program code in Thurlo is operative to store data in *any* of the memory devices of the type described in the driver — *not only* in the memory device that stores the driver.

In contrast, independent Claims 15, 33, 37, 39, 88, and 91 recite that the program code is operative to store data *only* in the memory device that stores the program code (and not in other memory devices of the same type). As discussed in Applicants’ specification, so restricting the write operation can provide an incentive for a user to pay an additional license fee to obtain the right to write in other memory devices. As also discussed in Applicants’ specification, if the memory device is write-once, the amount of data that can be stored in the memory device is limited, thereby effectively limiting the use of the program code if the program code could write data *only* in the memory device.

In summary, Thurlo does not teach program code that is operative to store data *only* in the memory device that stores the program code. This provides an additional ground of patentability for independent Claims 15, 33, 37, 39, 88, and 91.

**C. Independent Claims 21, 24, 28, 33, 90, and 91**

Independent Claims 21, 24, 28, 33, 90, and 91 each recite an element relating to a memory device storing encrypted program code and an identifier. A host device can decrypt the encrypted program code using the identifier. In this way, the program code is “tied” to the memory device. That is, the program code can be decrypted on any host device as long as the program code is coming from the memory device (because the memory device provides the identifier needed to decrypt the program code along with the program code). If, however, the program code is copied to a different memory device, which would not have the same identifier, the host device would not be able to decrypt the program code because the different memory device would only provide the program code and not the identifier needed to decrypt the program code. The Office Action admitted that this element was not shown in Thurlo and relied upon U.S. Patent No. 6,141,756 to Bright et al. and U.S. Patent No. 6,308,317 to Wilkinson et al. in an attempt to cure this deficiency. However, one skilled in the art would not have been motivated to make the proposed combination because it would change the basic operating principle of Bright et al.

Bright et al. is concerned about a processor executing an untrusted program from an external source. To ensure that only trusted programs are executed, Bright et al. describes a system in which trusted programs are encrypted with a key that is embedded in the processor. Since only trusted sources will have knowledge of the key, the processor can be assured that any program that is decrypted by the processor’s key is from a trusted source and, therefore, safe to execute. The Office Action admitted that this teaching is insufficient to yield the claimed inventions because Bright et al. stores its identifier in the processor and not in the memory device that provides the program code, as recited in independent Claims 21, 24, 28, 33, 90, and 91. In

an attempt to cure this deficiency, the Office Action relied upon Wilkinson et al., which purportedly teaches encrypting an application on a card with an identifier stored on the card. However, one skilled in the art would not have been motivated to make this combination because moving the key from the processor to the memory device would change the basic operating principle of Bright et al.

As discussed above, by using a key embedded in the processor, the processor in Bright et al. knows that any program that can be decrypted using the key is from a trusted source and is, therefore, safe to execute. Under the proposed combination, the key would be moved from the processor to the memory device that provides the program. This eliminates the processor's ability to "check" the trustworthiness of the program and creates a "fox guarding the hen house" situation by delegating security (the key) to the very entity that Bright et al. is trying to guard against (the provider of the program code). Instead of the processor knowing that a program is safe because it is able to unlock the program with the processor's key, the processor would have no assurances that the provided program is safe. This reintroduces the very problem that Bright et al. sought to overcome.

Because the proposed combination would change the basic operating principle of Bright et al., one skilled in the art would not have been motivated to make the proposed combination. This provides an additional ground of patentability for independent Claims 15, 33, 37, 39, 88, and 91 and their dependent claims.

### **III. Claim Objection**

Claim 91 was objected to as being a duplicate of Claim 28. Applicants respectfully submit that it is Claim 90 — not Claim 91 — that is a duplicate of Claim 28. In this Amendment, Applicants have amended Claim 90 in accordance with the agreements reached

during the telephone interview on October 26, 2006, which is referenced in the Interview Summary attached to the Office Action.

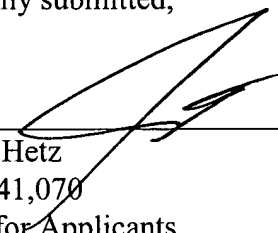
#### **IV. Conclusion**

In view of the foregoing amendments and remarks, Applicants respectfully submit that this application is in condition for allowance. Reconsideration is respectfully submitted. It should be noted that while only some elements of the independent claims were discussed above, other elements of the independent claims, as well as the dependent claims, provide additional grounds of patentability. Applicants reserve the right to present these additional grounds at a later time, if necessary.

If there are any questions concerning this Amendment, the Examiner is invited to contact the undersigned attorney at (312) 321-4719.

Dated: February 5, 2007

Respectfully submitted,



\_\_\_\_\_  
Joseph F. Hetz  
Reg. No. 41,070  
Attorney for Applicants

BRINKS HOFER  
GILSON & LIONE  
P.O. Box 10395  
Chicago, Illinois 60610  
(312) 321-4719